LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-PROC-464065

MIDAS (Material Implementation, Database, and Analysis Source): A comprehensive resource of material properties

M. Tang, P. Norquist, N. Barton, K. Durrenberger , J. Florando, A. Attia

December 15, 2010

NECDC 2010
Los Alamos, NM, United States
October 18, 2010 through October 22, 2010

Disclaimer

MIDAS (Material Implementation, Database, and Analysis Source):

A comprehensive resource of material properties

Meijie Tang, Peter Norquist, Nathan Barton, Kevin Durrenberger, Jeff Florando, Armand Attia

Lawrence Livermore National Laboratory, Livermore, CA 94551

MIDAS is aimed to be an easy-to-use and comprehensive common source for material properties including both experimental data and models and their parameters. At LLNL, we will develop MIDAS to be the central repository for material strength related data and models with the long-term goal to encompass other material properties. MIDAS will allow the users to upload experimental data and updated models, to view and read materials data and references, to manipulate models and their parameters, and to serve as the central location for the application codes to access the continuously growing model source codes. MIDAS contains a suite of interoperable tools and utilizes components already existing at LLNL: MSD (material strength database), MatProp (database of materials properties files), and MSlib (library of material model source codes). MIDAS requires significant development of the computer science framework for the interfaces between different components. We present the current status of MIDAS and its future development in this paper.

Introduction to MIDAS (Material Implementation, Database, and Analysis Source)

Materials modeling and materials research are at the cutting edge at DOE labs and other organizations. With experiments planned at gas guns, HEAF, Z-pinch, and NIF, new experimental data are expected to provide new and/or updated information about materials properties. Meanwhile, new model developments have produced new model source codes that include implementation of complex models such as the latest multi-scale physics based strength models [1]. While many old and new experimental results exist, there is no common database repository to organize them. On the other hand, materials properties used in simulation codes are hard to come by, and comparing model fits to experimental results is difficult. Traditionally, the 'bluebook' contains the standard strength and EOS models and parameters [2]. However, the need to include new materials, updated calibration with new experimental data and new model fits is often necessary. Few tools exist to assist analysis in determining the best parameters for a given regime of conditions (e.g., ambient, shock, plasma, etc). In addition, material model implementations are sometimes shared, but no common source code repository exists to ensure code developers have access to the latest additions. Furthermore, each code has to build its own interface to access the materials library and database. MIDAS aims to provide a unified solution to all of the above problems by supplying the computer science framework necessary to organize and house all experimental data, models, parameters and model source codes.

A few years ago, Rob Neely et al. initiated the MIDAS project [3]. It intended to utilize three already existing components (MSD, MatProp, MSlib) at LLNL. It produced a prototype application MSD GUI to view materials strength data and model fits [4]. It also created a limited set of physics models and parameters within the MatProp database[5]. However, at that time, there was not enough resources to develop the interfaces between different components. Recently, we revived the MIDAS project with the strong goal to integrate the different parts to develop an easy-to-use and comprehensive resource to build a materials property database, to view experimental data, model fits, references, as well as to enable application codes to use the latest and well documented models and parameters. When fully developed, MIDAS will provide a powerful application and framework for storage, management, and deployment of material properties. We focus on the material strength property at the current stage of development. In this paper, we will explain each component of MIDAS with its significant development plans, and summarize various MIDAS products to be developed at the end.

Materials Strength Database (MSD)

The MSD is an application allowing users to view and interact with experimental data and corresponding material strength models [4]. In MSD, the old prototype was a stand-alone JAVA-based application GUI that was coded with the physics models and associated user-interactive features. Material data and strength model parameters were imported from a remote SQL (Structured Query Language) database or locally from data files in XML (eXtensible Markup Language) format.

Currently, approximately thirty (30) material datasets and their associated strength model parameter sets are available through the MSD application GUI. A partial list of materials and material models supported currently (see Table I) are provided below. Experimental strength data (stress-strain curves) for individual materials can be plotted

together with various model fits, to assess the goodness-of-fit. Users can further change model parameter values to evaluate the effect of perturbations on the resulting fit-to-data. Users can also test the sensitivity of model fits to various parameters in the models. Additional plot types (e.g., strength vs. temperature and strength vs. strain-rate) are available to provide the user with more insight into rate and temperature dependent material behavior. The old prototype MSD GUI was a portable application. It used its own hard-coded physics models and called its own database for materials properties, and had no interface with MSlib or MatProp.

In the new development, MSD is a client-server Web 2.0 application allowing users to view and interact with experimental data, physics models and their parameters. It also allows users to upload experimental data and model fits, contributing to the model parameter and experimental data database (after verification and classification). The MSD client-server system is access controlled by means of LLNL LDAP servers. The MSD system consists of the following components: a Web 2.0 application GUI front end, called MatScipedia, which provides experimental data and model output plotting capability which runs in any modern web browser, an Apache web server and Tomcat Web Container, and a MySQL relational database server. To create the model plots, calls to MSlib at the same server are made which return computed numerical data to be plotted in the browser. See Fig. 1 for the schematic drawing of the MSD setup. MSD also contains a suite of utilities for uploading experimental data, model and/or model parameters, adding to the continuously growing database, through MatScipedia and a separate JAVA-based transportable data collector. This database will progressively produce MatProp sets of material model parameter XML files, which contain the latest inclusion of model fits and well documented previous model fits ready to be used directly as inputs to simulation codes.

Some additional features are possible in MSD. For example, one can import a hydro-code input deck, extract the strength model parameters for each defined material, view and/or manipulate the material model parameters within MatScipedia, and allow exporting of the parameters in an alternate format.

Material Strength Library (MSlib)

MSlib is a shared constitutive model library containing verified coding of validated models [6]. Many of these models are decomposed into EOS, strength, shear modulus, yield surface, and failure components. This decomposition enhances interoperability and provides compartmentalization of expertise for model developers. MSlib also includes models that are more integrated due to stronger cross-couplings such as the Gurson and TEnsile PLAsticity (TEPLA) models, and the latest multiscale physics based strength models. Recent improvements to MSlib have enhanced portability: as a stand-alone library the code now resides in its own source code repository managed by LLNL's sourceforge, and host code EOS callback capabilities have been added.

In the new MIDAS development, MSlib will remake a more versatile API to interface with various application codes at LLNL. It will utilize the Babel tool to develop an outer layer wrapper interface that will facilitate the interaction of MSlib with application codes written in various computer languages such as C, C++, Fortran, Python etc. By doing so, it will develop a framework for assessing materials model parameters and reduce code duplication. In the new development, MSlib will internally call MatProp extract material parameters through a MatProp object. Currently, the host code must

parse material input parameters and populate data structures for MSlib. By moving this capability into MSlib, based on MatProp, material inputs will be unified across host codes and it will be possible to implement material model components with less modification to the host code. For certain models, host codes will likely maintain their native material parameter input capabilities.

Incorporation of MatProp capabilities into MSlib will also enable the development of a stand-alone driver program for performing simple simulations using MSlib material models. This stand-alone driver will be able to simulate experiments such as compression tests and Hopkinson pressure bar tests used in calibrating and assessing material models. This stand-alone driver will then serve as a back-end for the web browser MatScipedia allowing users to interact with the material models and calibrate parameters. In this way, users will exercise the same code (MSlib) using the same parameters (through MatProp) in both the web interface and the production codes. This will help to reduce inconsistencies and input errors, although code versioning and consistent documentation will be important.

MatProp Database and Access Library

The MatProp consists of three parts: the access library codes in C++, the MatProp database files in XML format, and the auxiliary separate JAVA utility to edit/change/export parameters in a specific XML data file in order for users to make up own input deck based on the default values. The MatProp database contains parameters for many materials and models, and includes material parameters in the following categories – flow stress, shear modulus, strength, damage, and equation of state (see Fig. 2). Currently, MatProp database contains parameters for Steinberg-Guinan and Steinberg-Lund constitutive models for metals, JWL equations of state for high explosives and Mie-Gruneissen equation of states for metals. There are also limited numbers of Johnson-Cook equation of state and MTS constitutive models. The model names and parameter naming conventions are defined in Ref. [7].

The database is stored as XML (eXtensible Markup Language) and has an access library written in C++. XML was chosen as a storage format to provide easy portability, as it is platform independent ASCII text and does not require any specific tools to read or access directly. It can be opened and edited directly as text or with the access library. The XML format was designed to allow the addition of any future parameters, models or materials, without having to allow for "special cases". It has a very simple versioning scheme which allows a value of a parameter to be versioned thus reducing the size of the files. This allows the user to retrieve the values of different models based upon a version for the values. If the version does not exist it will simply return the default value. An important new feature is to tie the version name with a person who manages the model/parameter set when the users need to know the development history and the background of the model. Future versions of the MatProp database and the access library will include the ability to read and write material descriptions that can be exchanged between codes. It will also include an ability to do unit conversions, i.e. have the library return parameters in user-specified unit systems.

Summary

When MIDAS is fully developed, it will provide a resource for materials properties storage, management, and deployment. The main products developed by MIDAS will include two materials database (one contains experimental data and models

and resides at the server, the second one contains models and their blessed sets of parameters in MatProp data formats and is releasable), the web browser MatScipedia for uploading/viewing/exporting data and model parameters, and a versatile MSlib API facilitating application codes to access the comprehensive and updated model source codes with well documented material parameters. To help visualize the interoperation between different components of the MIDAS project, the architecture of MIDAS is shown in Fig. 3, where three main functions are highlighted. The blue oval outlines the interfaces for the capability to build-up the developers' database containing both experimental data and model parameters and a releasable MatProp database containing models and their blessed parameter sets. The build-up of the database starts from experimental data input through the MatScipedia browser. Once uploaded to the system, it will be reviewed and properly categorized, then subsequently stored into the permanent database with thorough documentations. Similarly, newly derived models or updated model parameters will be uploaded and stored. After verification and assessment, improved model and parameter sets will be written into the releasable MatProp database and create a new version of MatProp database. The green oval outlines the interfaces for the web users to view experimental data, model fits, and manipulate model parameters. User requests sent over the browser will call command line drivers to the model source code MSlib, which then internally accesses the model parameters in MatProp to send back numerical data to be plotted and displayed at the browser. The red oval outlines the interfaces for application codes to access the same model codes and parameters through MSlib and MatProp. The Babelized interface of MSlib will allow it to be called by the application codes written in various programming languages (C, C++, Python, Fortran).

While the computer science framework discussed in this paper is the backbone of the MIDAS project, an important application goal of MIDAS is to facilitate the development of improved strength models of materials. It is crucial for MIDAS to have access to the latest and best quality experimental data. For this purpose, we want to engage with the broad scientific community to obtain the data, and subsequently to produce models based on the data through direct numerical fitting or more complex simulations. In addition, MIDAS will house strength models with proven record of producing physically sound results through various applications and provided by users.

Finally, the MIDAS vision of building an easy-to-use and comprehensive resource for material properties is shared not only by the DOE labs, but also by the DOD and other organizations. We will initially focus on internal and tri-lab applications for its usage. We certainly plan to make MIDAS products partially available for other applications outside LLNL. We will first focus on materials strength response. The computer science infrastructure to be built can be expanded in the future to interface with other database already existing, and to integrate other materials properties such as equation of state, fracture, failure, irradiation, and high explosives properties etc.

Acknowledgements

This work is performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory

under Contract DE-AC52 07NA27344. We acknowledge the support and advice from Brad Wallin and Adam Schwartz. We also acknowledge the informative discussions with Paul Amala, Grant Bazan, Rob Cavallo, Dana Goto, Burl Hall, John Klepeis, Rose McCallen, Rob Neely, Daniel Orlikowski, Bruce Watkins, and Christine Wu.

References

1. R. Becker, A. Arsenlis, G. Hommes, J. Marian, M. Rhee, and L. Yang, “A tantalum strength model using a multiscale approach: version 2”, LLNL-TR-417075, September, 2009.

2. Steinberg, D. J., “Equation of state and strength properties of selected materials”, UCRL-MA-106439 (1996).

3. Rob Neely, Dana Goto, Richard Becker, and Albert Nichols, “Material Implementation, Database and Analysis Source (MIDAS)”, LLNL-PRES-409545, 2008.

4. Peter Norquist, Dana Goto, and Tien Shen, “MSD: Materials Strength Database”, UCRL-POST-200663, NECDC 2004.

5. Kevin Durrenberger, Rob Neely, Dana Goto, “MatProp: An Extensible Material Properties Database and Software Access Library”, LLNL-POST-407578, NECDC 2008.

6. Rich Becker, Paul Amala, Armand Attia, Grant Bazan, Burl Hall, Rob Neely, Paul Renard, "Sharing MSlib material model library: Evaluation and Recommendations," ASC IC L2 Milestone Review, June 23, 2008.

7. J. K. Durrenberger, R. C. Becker, D. M. Goto, J. R. Neely, B. K. Walin, “MatProps: Material Properties Database and Associated Access Library”, UCRL-TR-233699, 2007.

Material	Strength Data	Steinberg-Guinan	Steinberg-Lund	Johnson-Cook	Zerilli-Armstrong
Aluminum (AA1100)	Y	Y	Y	Y	---
Aluminum (AA6061)	Y	Y	---	Y	---
Aluminum (AA7075)	Y	Y	Y	Y	---
Beryllium	Y	Y	Y	---	Y (2 avail.)
Copper (soft-annealed)	Y	Y	---	Y	Y
Copper (half-hard)	Y	Y (2 avail.)	---	---	---
Iron	Y	Y	Y	Y	Y
Magnesium (AZ-31B)	---	Y	---	---	---
Molybdenum	Y	Y	Y	Y	Y
Nickel	Y	Y	Y	Y	---
Stainless Steel (21-6-9, Nitronic-40)	Y	Y	Y	Y	Y
Stainless Steel (304)	Y	Y	Y	---	---
Steel (4340)	Y	Y	Y (2 avail.)	Y (2 avail.)	Y
Steel (AerMet-100 soft-annealed)	Y	Y	Y	Y	---
Steel (AerMet-100 HRC 55)	Y	Y	Y	Y	---
Steel (HSLA-100)	Y	Y	Y	Y	Y
Steel (HY-100)	Y	Y	Y	Y	Y
Steel (HY-80)	Y	Y	Y	Y	Y
Steel (Rolled Homogeneous Armor)	Y	Y	Y	Y	Y
Tantalum	Y	Y	Y	Y (2 avail.)	Y
Tantalum-10% Tungsten	Y	Y	Y	Y	Y
Titanium	Y	Y	Y (3 avail.)	Y (3 avail.)	---
Titanium-6% Aluminum-4% Vanadium	Y	Y	Y	Y (3 avail.)	Y (2 avail.)
Tungsten	Y	Y	Y (2 avail.)	Y	---
Uranium-6% Niobium	Y	Y	Y	Y	---
Uranium	Y	Y	Y (2 avail.)	---	---
Uranium-0.75% Titanium	Y	Y	Y	Y	---
Vanadium	Y	Y	Y	---	---

Table 1. A partial list of material type and strength models implemented in the old MSD GUI [4].

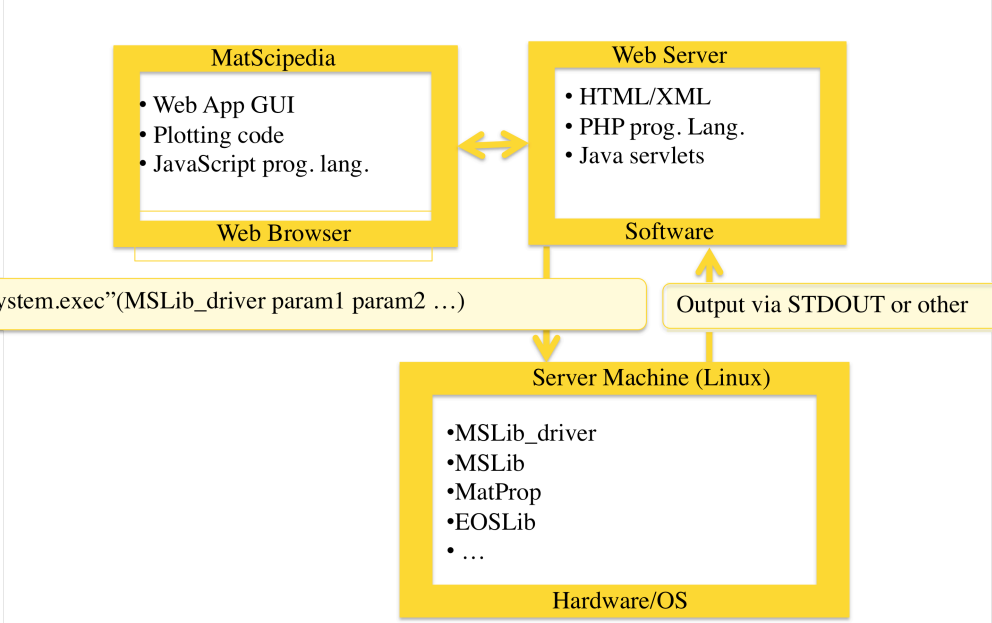

Fig. 1. A schematic drawing of MSD system set-up. It consists of the web browser (MatScipedia), the web server, and the necessary software to build the capabilities for browsing model and data through model source code (MSlib) and using documented material properties (MatProp).

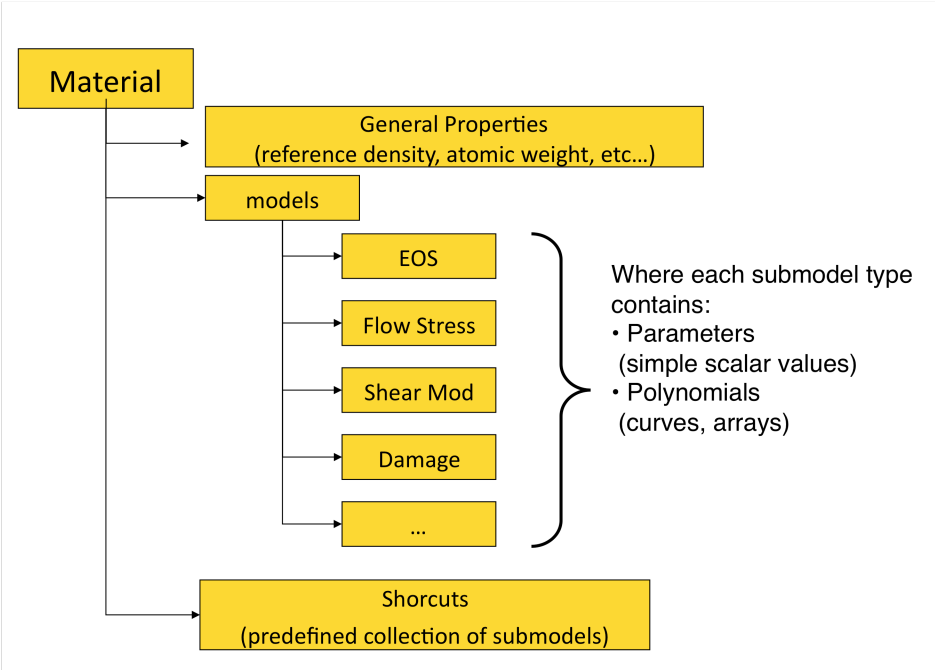

Fig. 2. The conceptual design of a MatProp material. The data formats are in XML and the access library is in C++.

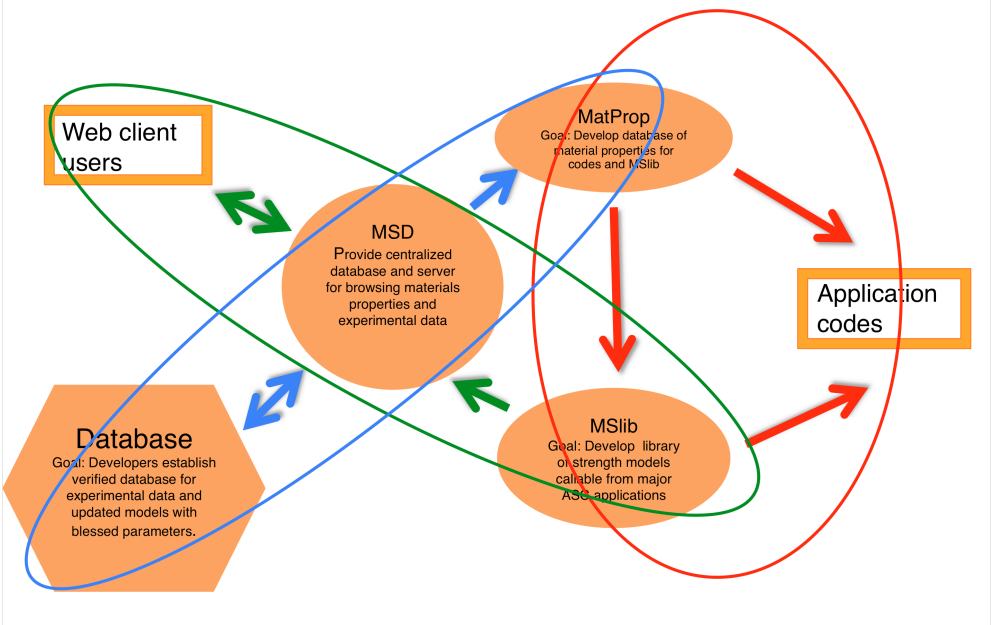

Fig. 3. The revived MIDAS architecture. It contains three main components: MSD, MatProp, and MSlib. When MIDAS is fully developed, it will provide three main capabilities: (1) (blue oval) Build-up database containing experimental data and blessed model fits and produce releasable MatProp data sets containing blessed model fits; (2) (green oval) Deploy web browser to enable users to upload/view/export experimental data, model parameters, and documentations; (3) (red oval) Enable application codes to directly access the model source codes with internal calls to model parameters.